

[illegible]

[illegible]

(2) 49  
(3) 78

DECLARATIONS  
CONVERT CLI FORMS AND CHARACTERISTICS

```

0000 1      .TITLE CNVCLIFRM - CONVERT COMMAND INPUT FORMS TYPE
0000 2      .IDENT 'V04-000'
0000 3
0000 4
0000 5 *****
0000 6 *****
0000 7      *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8      *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9      *  ALL RIGHTS RESERVED.
0000 10
0000 11      *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12      *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13      *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14      *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15      *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16      *  TRANSFERRED.
0000 17
0000 18      *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19      *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20      *  CORPORATION.
0000 21
0000 22      *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23      *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24
0000 25 *****
0000 26 *****
0000 27
0000 28
0000 29      ++
0000 30      FACILITY:      CLI UTILITIES
0000 31
0000 32      ABSTRACT:      THIS ROUTINE CONVERTS A FORMS/CHARACTERISTIC TYPE
0000 33                      INTO A BINARY VALUE.
0000 34
0000 35
0000 36      ENVIRONMENT:  NATIVE MODE, NON-PRIVILEGED
0000 37
0000 38      AUTHOR:      STEVE BECKHARDT,  CREATION DATE:  13-FEB-78
0000 39
0000 40      MODIFIED BY:
0000 41
0000 42      V03      GWF0107      GARY FOWLER      21-SEP-1981
0000 43                      Change logical name used in opening files to SYSSMANAGER
0000 44
0000 45      V02      LMK0001      LEN KAWELL      15-FEB-1980
0000 46                      ADD DE-QUEUE CHARACTERISTICS CONVERSION.
0000 47      --

```



```

0000 49 .SBTTL DECLARATIONS
0000 50 :
0000 51 : INCLUDE FILES:
0000 52 :
0000 53 :
0000 54 : MACROS:
0000 55 :
0000 56 :
0000 57 :
0000 58 : EQUATED SYMBOLS:
0000 59 :
0000 60 $CLIMSGDEF ; DEFINE CLI ERROR CODES
0000 61 $FABDEF ; FAB DEFINITIONS
0000 62 $RABDEF ; RAB DEFINITIONS
0000 63
00000087 0000 64 RCBFSZ=135 ; RECORD BUFFER SIZE FOR FORMS
0000 65 ; DEFINITION FILE
0000 66
0000 67 :
0000 68 : OWN STORAGE:
0000 69 :
0000 70 PURE_SECTION PURE_CLIUTL
0000 71
0000 72
52 45 47 41 4E 41 4D 24 53 59 53 00' 0000 73 FFILNAM: .ASCIC /SYSSMANAGER:FORMSTYPE.DAT/ ; FILE NAME OF FORMS DEFINITION FILE
44 2E 45 50 59 54 53 4D 52 4F 46 3A 000C
54 41 0018
19 0000
52 45 47 41 4E 41 4D 24 53 59 53 00' 001A 74 CFILNAM: .ASCIC /SYSSMANAGER:CHARTYPE.DAT/ ; FILE NAME OF CHARACTERISTICS DEFIN
41 44 2E 45 50 59 54 52 41 48 43 3A 0026
54 0032
18 001A
0033 75
0033 76

```

```

0033 78      .SBTTL  CONVERT CLI FORMS AND CHARACTERISTICS
0033 79      :++
0033 80      : FUNCTIONAL DESCRIPTION:
0033 81      :
0033 82      : THESE ROUTINES ARE CALLED TO CONVERT THE FORMS/CHARACTERISTICS
0033 83      : ENTERED ON A COMMAND INTO A BINARY BYTE VALUE.
0033 84      : THIS WILL CONVERT EITHER A NUMBER DIRECTLY, OR
0033 85      : A ALPHANUMERIC NAME.  ALPHANUMERIC FORMS NAMES ARE
0033 86      : STORED IN FILE "SYSSMANAGER:FORMSTYPE.DAT" AND CHARACTERISTICS
0033 87      : NAMES ARE STORED IN "SYSSMANAGER:CHARTYPE.DAT".
0033 88      : FORMS/CHARACTERISTICS DEFINITIONS IN THE FILE MUST BE IN
0033 89      : THE FOLLOWING FORMAT:
0033 90      : % TYPE NUMBER
0033 91      : THE PERCENT SIGN MUST BE THE FIRST CHARACTER IN THE RECORD.
0033 92      :
0033 93      : CALLING SEQUENCE:
0033 94      : VIA "CALL" INSTRUCTION
0033 95      :
0033 96      : INPUT PARAMETERS:
0033 97      :
0033 98      : 4(AP) IS THE ADDRESS OF A QUAD WORD DESCRIPTOR FOR THE TYPE CODE
0033 99      : 8(AP) IS THE ADDRESS OF BYTE TO STORE RESULT(OPTION)
0033 100     :
0033 101     : IMPLICIT INPUTS:
0033 102     :
0033 103     : NONE
0033 104     :
0033 105     : OUTPUT PARAMETERS:
0033 106     :
0033 107     : THE CONVERTED FORMS/CHAR TYPE IS RETURNED IN R1 AND AT THE LOCATION
0033 108     : SPECIFIED IF THE CALL ARGUMENT LIST HAD MORE THAN ONE ARGUMENT.
0033 109     :
0033 110     : IMPLICIT OUTPUTS:
0033 111     :
0033 112     : CONVERTED VALUE IN R1
0033 113     :
0033 114     : COMPLETION CODES:
0033 115     :
0033 116     : R0 IS SET TRUE OR FALSE DEPENDING UPON SUCCESS OR FAILURE OF CONVERSION
0033 117     :
0033 118     : SIDE EFFECTS:
0033 119     :
0033 120     : NONE
0033 121     :
0033 122     : --
0033 123     :
0033 124     : PURE_SECTION  PURE_CLIUTL
0033 125     :
0033 126     : LIB$CNVCLIFORMS:: : CONVERT FORMS TYPE
0033 127     : .WORD  *M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
0033 128     : MOVAB  FFILNAM,R6 : SET ADDRESS OF FORMS FILE
0033 129     : BRB    CONVERT :
0033 130     :
0033 131     : LIB$CNVCLIQCHAR:: : CONVERT CHARACTERISTICS TYPE
0033 132     : .WORD  *M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
0033 133     : MOVAB  CFILNAM,R6 : SET ADDRESS OF CHAR FILENAME
0041 134

```

56 CB AF OFFC  
06 9E 11

56 DA AF OFFC  
9E



```
00000000'EF 04 BC 7F 0041 135 CONVERT:
03 01 FB 0041 136 ; FIRST SEE IF WE CAN CONVERT A NUMBER
00AC 31 0044 137 ; POINT TO TOKEN DESCRIPTOR
5A 04 BC 7D 004B 138 ; TRY TO CONVERT IT
0051 139 ; CONVERSION FAILED
0051 140 ; SUCCESS! VALUE IN R1
0055 141
0055 142 5$: ; NUMBER CONVERSION FAILED. TRY LOOKING IT UP IN FILE
0055 143 MOVQ @4(AP),R10 ; PUT TOKEN DESCRIPTOR IN R10,R11
0055 144
0055 145 ; ALLOCATE SPACE FOR FAB, RAB, AND BUFFERS ON STACK
57 B0 AE 9E 0055 146 MOVAB -FAB$C_BLN(SP),R7 ; R7 POINTS TO FAB
58 BC A7 9E 0059 147 MOVAB -RAB$C_BLN(R7),R8 ; R8 POINTS TO RAB
59 FF78 C8 9E 005D 148 MOVAB -<RCBF5Z+1>(R8),R9 ; R9 POINTS TO RECORD BUFFER
5E 59 D0 0062 149 MOVL R9,SP ; SET STACK POINTER
0065 150
68 0094 8F 00 68 00 2C 0065 151 ; CLEAR FAB AND RAB
0065 152 MOVCS #0,(R8),#0,#RAB$C_BLN+FAB$C_BLN,(R8)
006D 153
006D 154 ; SET VARIOUS FAB FIELDS
67 5003 8F B0 006D 155 ASSUME FAB$B_BLN EQ FAB$B_BID+1
16 A7 02 90 006D 156 MOVW #<FAB$C_BLN@8>+FAB$C_BID,FAB$B_BID(R7) ; BLOCK ID AND LENGTH
34 A7 86 90 0072 157 MOVW #FAB$M_GET,FAB$B_FACT(R7) ; FILE ACCESS
2C A7 66 9E 0076 158 MOVW (R6)+,FAB$B_FNS(R7) ; FILE NAME SIZE
007A 159 MOVAB (R6),FAB$B_FNA(R7) ; FILE NAME ADDRESS
007E 160
007E 161 ; OPEN FILE
007E 162 $OPEN FAB=(R7)
03 50 E8 0087 163 BLBS R0,15$
008B 31 008A 164 10$: BRW R0,15$ ; SUCCESSFUL OPEN
008D 165 ; ERROR (ASSUME FILE NOT FOUND)
008D 166 15$:
008D 167 ; SET VARIOUS RAB FIELDS
68 4401 8F B0 008D 168 ASSUME RAB$B_BLN EQ RAB$B_BID+1
3C A8 67 9E 0092 169 MOVW #<RAB$C_BLN@8>+RAB$C_BID,RAB$B_BID(R8) ; BLOCK ID AND LENGTH
1E A8 00 90 0096 170 MOVAB (R7),RAB$B_FAB(R8) ; FAB ADDRESS
24 A8 69 9E 009A 171 MOVW #RAB$C_SEQ,RAB$B_RAC(R8) ; RECORD ACCESS
20 A8 0087 8F B0 009E 172 MOVAB (R9),RAB$B_UBF(R8) ; USER BUFFER ADDRESS (ON STACK)
009E 173 MOVW #RCBF5Z,RAB$B_USZ(R8) ; USER BUFFER SIZE
00A4 174
00A4 175 $CONNECT RAB=(R8)
5F 50 E9 00AD 176 BLBC R0,70$ ; CONNECT ERROR
00B0 177 20$:
00B0 178 ; READ A LINE FROM FILE
53 50 E9 00B9 179 $GET RAB=(R8)
00BC 180 BLBC R0,70$ ; ERROR (ASSUME END OF FILE)
00BC 181
00BC 182 ; GET LENGTH OF RECORD AND STORE A 0 AT THE END (FOR PARSING ROUTINES)
50 22 A8 3C 00BC 182 MOVZWL RAB$W_RSZ(R8),R0 ; RECORD SIZE
6940 94 00C0 183 CLRB (R9)[R0] ; WE NOW HAVE AN ASCIZ STRING
00C3 184
00C3 185 ; IS THIS LINE INTERESTING? (DOES IT BEGIN WITH %)
56 59 D0 00C3 186 MOVL R9,R6 ; POINTER TO BUFFER
25 86 91 00C6 187 CMPB (R6)+,#^A/%/
E5 12 00C9 188 BNEQ 20$ ; NOT OF INTEREST. GET NEXT LINE
00CB 189
00CB 190 ; NOW GET FIRST TOKEN ON LINE AND SEE IF IT MATCHES INPUT SYMBOL
FF32' 30 00CB 191 BSBW CHR$GETOKEN ; RETURNS TOKEN DESCRIPTOR IN R3,R4
```

```
64 53 00 6B E0 13 00CE 192 BEQL 20$ ; NULL TOKEN; GET NEXT LINE
5A 2D 00D0 193 CMPC5 R10,(R11),#0,R3,(R4) ; IT IS CONSIDERED A MATCH IF THE INPUT
50 B5 00D6 194 TSTW R0 ; SYMBOL IS AN INITIAL SUBSTRING OF THE
; TOKEN (R0=0)
D6 12 00D8 195 BNEQ 20$ ; NO MATCH; GET NEXT LINE
; HAVE A MATCH. R6 POINTS TO NEXT TOKEN (HOPEFULLY VALUE)
FF23' 30 00DA 198 BSBW CHR$GETOKEN
30 13 00DA 199 BEQL 70$ ; NULL TOKEN --> ERROR
; HAVE TOKEN. SET UP TO CALL ROUTINE TO CONVERT TO BINARY
7E 53 7D 00DF 202 MOVQ R3,-(SP) ; PUT TOKEN DESCRIPTOR ON STACK
; BUILD ARGUMENT LIST ON STACK
00000000'EF 6E 7F 00E2 205 PUSHAQ (SP) ; POINTER TO TOKEN DESCRIPTOR
01 FB 00E4 206 CALLS #1,LIB$CVT_DECBIN ; CONVERT TO BINARY
8E 7C 00EB 207 CLRQ (SP)+ ; REMOVE DESCRIPTOR
1F 50 E9 00ED 208 BLBC R0,70$ ; CONVERSION ERROR
; SUCCESS RETURN (WITH CLOSE). VALUE IS IN R1
51 DD 00F0 211 50$: PUSHL R1 ; SAVE R1 ACROSS CLOSE
; CLOSE FILE
02 BA 00FB 214 POPR #^M<R1> ; RESTORE R1
; SUCCESS RETURN (WITHOUT CLOSE). VALUE IS IN R1.
50 00030001 8F D0 00FD 216 60$: MOVL #CLIS_NORMAL,R0 ; INDICATE SUCCESS
01 6C D1 0104 218 CMPL (AP),#1 ; STORE IN MEMORY IF MORE THAN ONE ARG
16 1B 0107 219 BLEQU 90$ ; DON'T STORE
08 BC 51 90 0109 220 MOVB R1,@8(AP) ; STORE
10 11 010D 221 BRB 90$
; FAILURE RETURN (WITH CLOSE)
010F 222 70$: $CLOSE FAB=(R7) ; CLOSE FILE
0118 225
; FAILURE RETURN (WITHOUT CLOSE)
50 00038832 8F D0 0118 226 80$: MOVL #CLIS_VALCNVERR,R0 ; INDICATE CONVERSION ERROR
04 011F 228 90$: RET
0120 230
0120 231
0120 232
0120 233 .END
```



CNVCLIFRM  
Symbol table

- CONVERT COMMAND INPUT FORMS TYPE C 6

15-SEP-1984 23:38:19 VAX/VMS Macro V04-00  
4-SEP-1984 23:15:15 [CLIUTL.SRC]CNVCLIFRM.MAR;1

Page 6  
(3)

\$\$TMP1	=	00000001		
\$\$TMP2	=	00000067		
CFILNAM		0000001A	R	02
CHRSGETOKEN		*****	X	02
CLIS_NORMAL	=	00030001		
CLIS_VALCNVERR	=	00038832		
CONVERT		00000041	R	02
FABSB_BID	=	00000000		
FABSB_BLN	=	00000001		
FABSB_FAC	=	00000016		
FABSB_FNS	=	00000034		
FABSC_BID	=	00000003		
FABSC_BLN	=	00000050		
FABSL_FNA	=	0000002C		
FABSM_GET	=	00000002		
FFILNAM		00000000	R	02
LIBSCNVCLIFORMS		00000033	RG	02
LIBSCNVCLIQCHAR		0000003B	RG	02
LIBSCVT_DECBIN		*****	X	02
RABSB_BID	=	00000000		
RABSB_BLN	=	00000001		
RABSB_RAC	=	0000001E		
RABSC_BID	=	00000001		
RABSC_BLN	=	00000044		
RABSC_SEQ	=	00000000		
RABSL_FAB	=	0000003C		
RABSL_UBF	=	00000024		
RABSW_RSZ	=	00000022		
RABSW_USZ	=	00000020		
RCBFSZ	=	00000087		
SYSSCLOSE	*****		GX	02
SYSSCONNECT	*****		GX	02
SYSSGET	*****		GX	02
SYSSOPEN	*****		GX	02

!-----!  
! Psect synopsis !  
!-----!

PSECT name	Allocation	PSECT No.	Attributes														
. ABS .	00000000 ( 0.)	00 ( 0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE				
\$ABSS	00000000 ( 0.)	01 ( 1.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE				
PURE_CLIUTL	00000120 ( 288.)	02 ( 2.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE				

!-----!  
! Performance indicators !  
!-----!

Phase	Page faults	CPU Time	Elapsed Time
Initialization	16	00:00:00.07	00:00:00.64
Command processing	109	00:00:00.81	00:00:06.07
Pass 1	200	00:00:05.06	00:00:19.86
Symbol table sort	0	00:00:00.50	00:00:02.49
Pass 2	62	00:00:01.04	00:00:03.70
Symbol table output	4	00:00:00.05	00:00:00.05

Psect synopsis output	2	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	395	00:00:07.57	00:00:32.85

The working set limit was 1200 pages.  
26652 bytes (53 pages) of virtual memory were used to buffer the intermediate code.  
There were 30 pages of symbol table space allocated to hold 466 non-local and 9 local symbols.  
233 source lines were read in Pass 1, producing 13 object records in Pass 2.  
18 pages of virtual memory were used to define 16 macros.

-----  
! Macro library statistics !  
-----

Macro library name	Macros defined
-----	-----
\$255\$DUA28:[CLIUTL.OBJ]CLIUTL.MLB;1	1
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
\$255\$DUA28:[SYSLIB]STARLET.MLB;2	12
TOTALS (all libraries)	13

590 GETS were required to define 13 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:CNVCLIFRM/OBJ=OBJ\$:CNVCLIFRM MSRC\$:CNVCLIFRM/UPDATE=(ENH\$:CNVCLIFRM)+EXECMLS/LIB+LIB\$:CLIUTL/LIB



0049 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

BCPR5DEF  
REQ

CVCLIBB  
LIS

INFO  
LIS

TYPE  
REQ

CHRSUB  
LIS

CVCLINUM  
LIS

SHODEVDEF  
REQ

CLIMAC  
MAR

CVCLIFRM  
LIS

DIGRAMS  
LIS

CALCMAX  
LIS

CLUTLMAC  
MAR

CUTTIME  
LIS

BCMDPRS  
LIS

SHOWDEF  
REQ

CREATE  
LIS